US 20210150674A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2021/0150674 A1**

CAI et al. (43) **Pub. Date: May 20, 2021**

(54) **TECHNIQUES FOR ROBUST IMAGE DENOISING**

(71) Applicant: **DISNEY ENTERPRISES, INC.**, Burbank, CA (US)

(72) Inventors: **Zhilin CAI**, Zurich (CH); **Tunc Ozan AYDIN**, Zurich (CH); **Marco MANZI**, Zurich (CH); **Ahmet Cengiz OZTIRELI**, Zurich (CH)

(21) Appl. No.: **16/795,486**

(22) Filed: **Feb. 19, 2020**

Related U.S. Application Data

(60) Provisional application No. 62/936,341, filed on Nov. 15, 2019.

Publication Classification

(51) **Int. Cl.**

G06T 5/00 (2006.01)

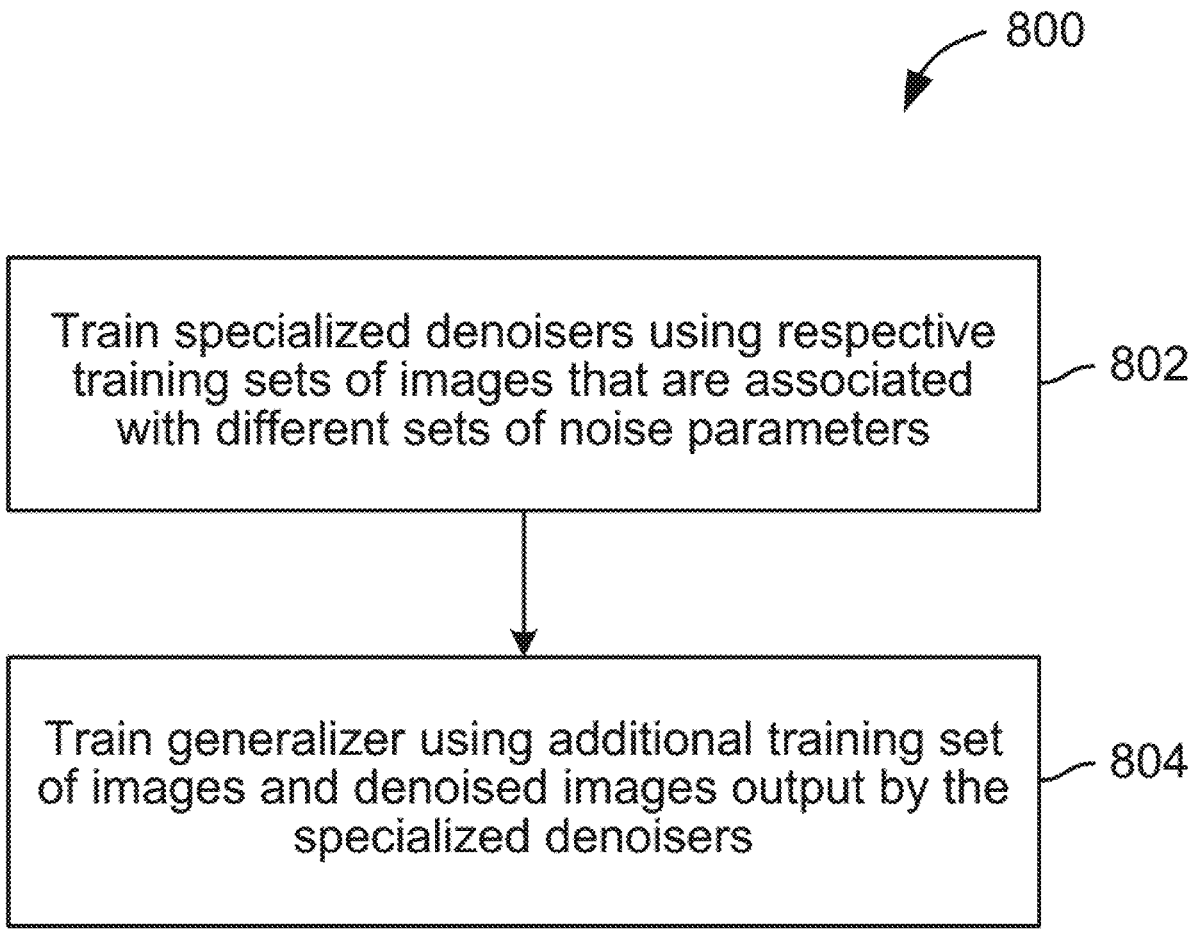
G06T 5/50 (2006.01)

(52) **U.S. Cl.**

CPC ***G06T 5/002*** (2013.01); ***G06T 5/50*** (2013.01); *G06T 2207/20221* (2013.01); *G06T 2207/20081* (2013.01); *G06T 2207/20084* (2013.01)

(57) **ABSTRACT**

Techniques are disclosed for training and applying a denoising model. The denoising model includes multiple specialized denoisers and a generalizer, each of which is a machine learning model. The specialized denoisers are trained to denoise images associated with specific ranges of noise parameters. The generalizer is trained to generate per-pixel denoising kernels for denoising images associated with arbitrary noise parameters using outputs of the specialized denoisers. Subsequent to training, a noisy image, such as a live-action image or a rendered image, can be denoised by inputting the noisy image into the specialized denoisers to obtain intermediate denoised images that are then input, along with the noisy image, into the generalizer to obtain per-pixel denoising kernels, which can be normalized and applied to denoise the noisy image.



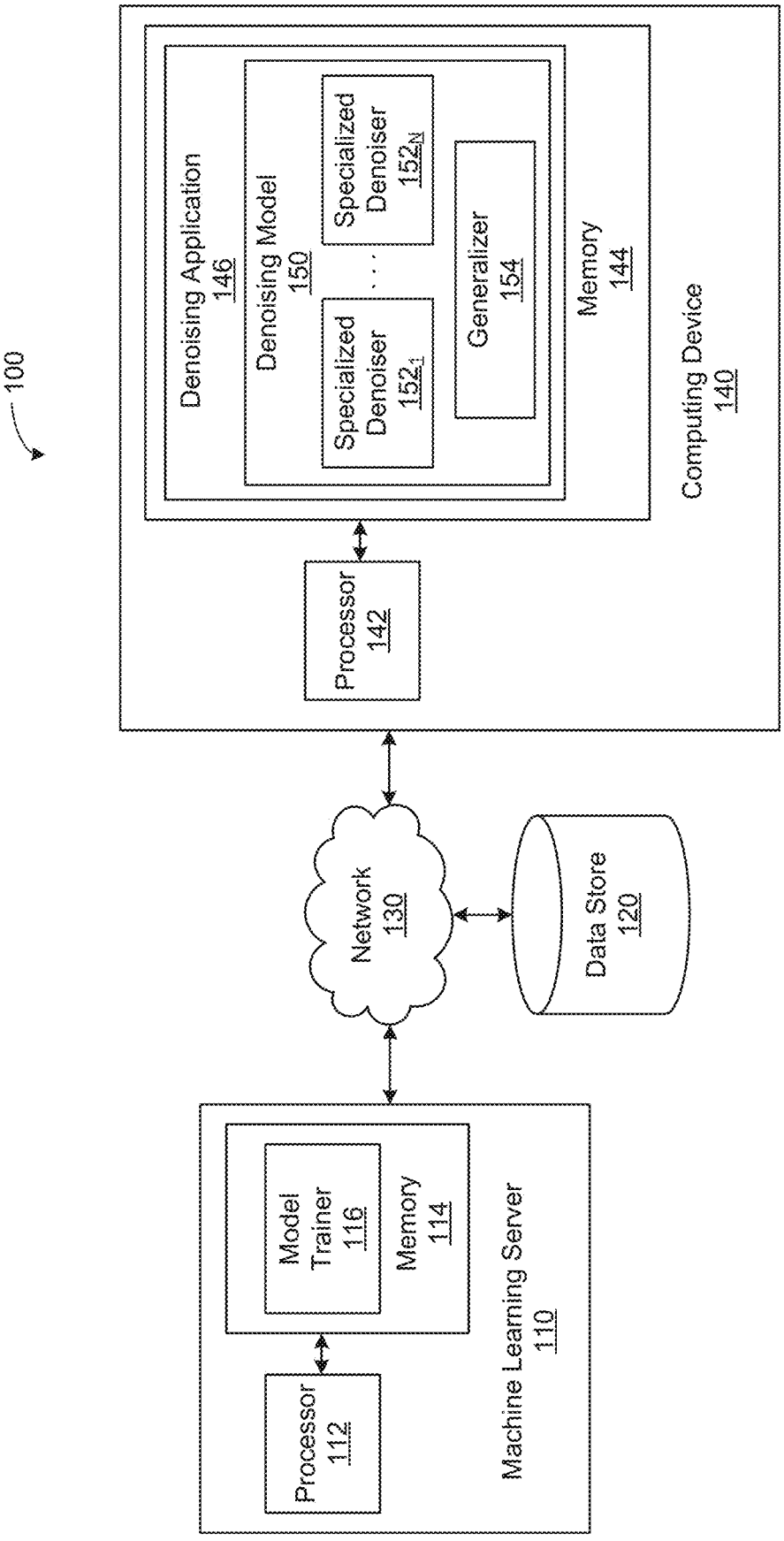

100
Machine Learning Server
110
Processor
112
Memory
114
Model
Trainer
116
Network
130
Data Store
120
Computing Device
140
Processor
142
Memory
144
Denoising Application
146
Denoising Model
150
Specialized
Denoiser
152₁
Specialized
Denoiser
152N
Generalizer
154

FIG. 1

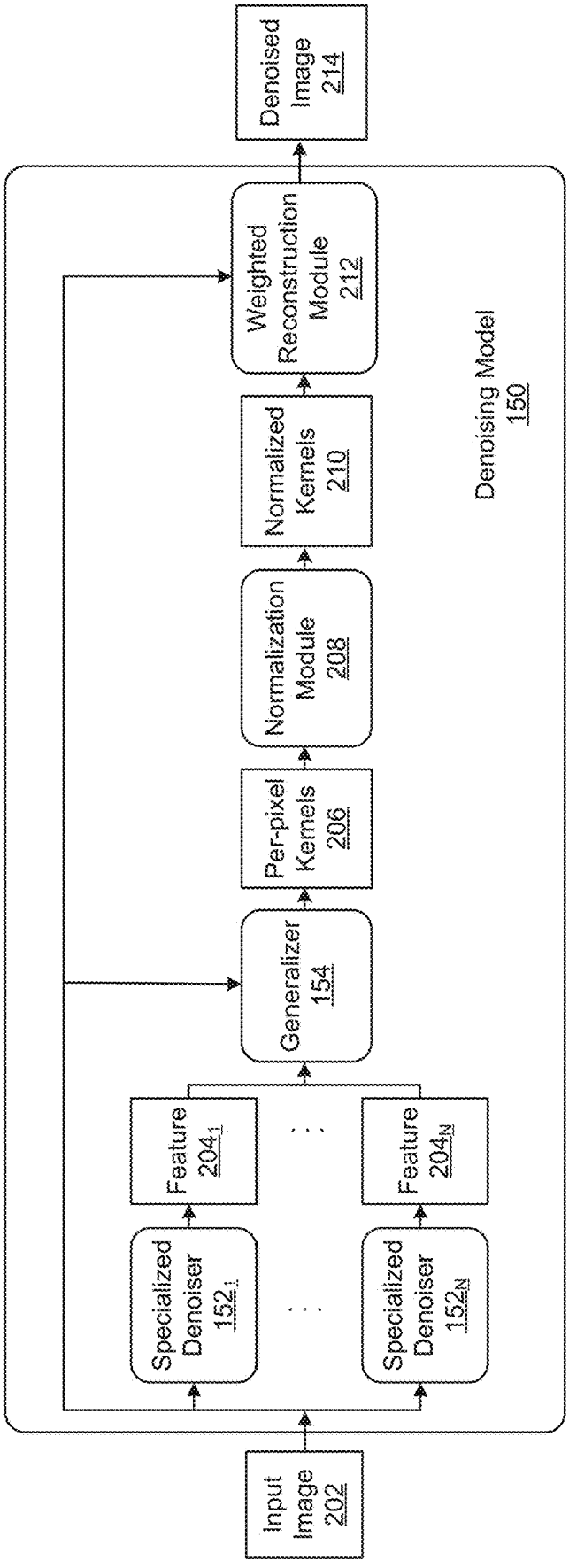

Input Image 202
Specialized Denoiser 152$_1$
Specialized Denoiser 152$_N$
Feature 204$_1$
Feature 204$_N$
Generalizer 154
Per-pixel Kernels 206
Normalization Module 208
Normalized Kernels 210
Weighted Reconstruction Module 212
Denoised Image 214
Denoising Model 150

FIG. 2

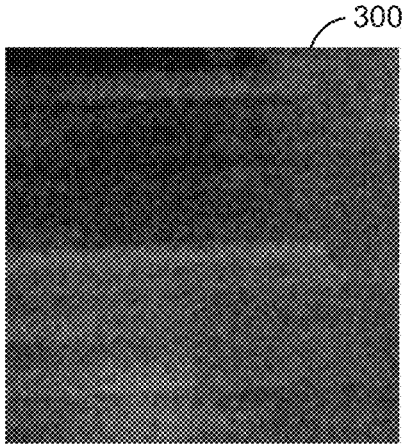


FIG. 3A

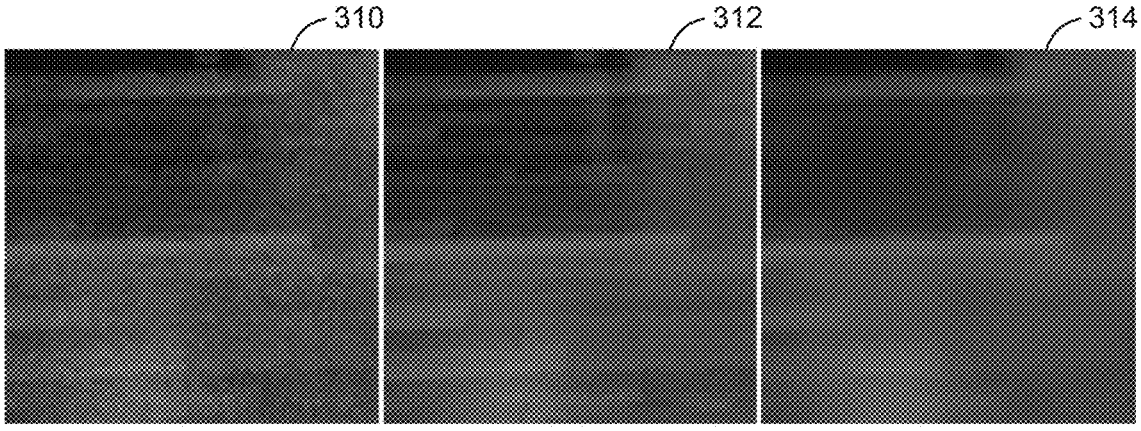


FIG. 3B

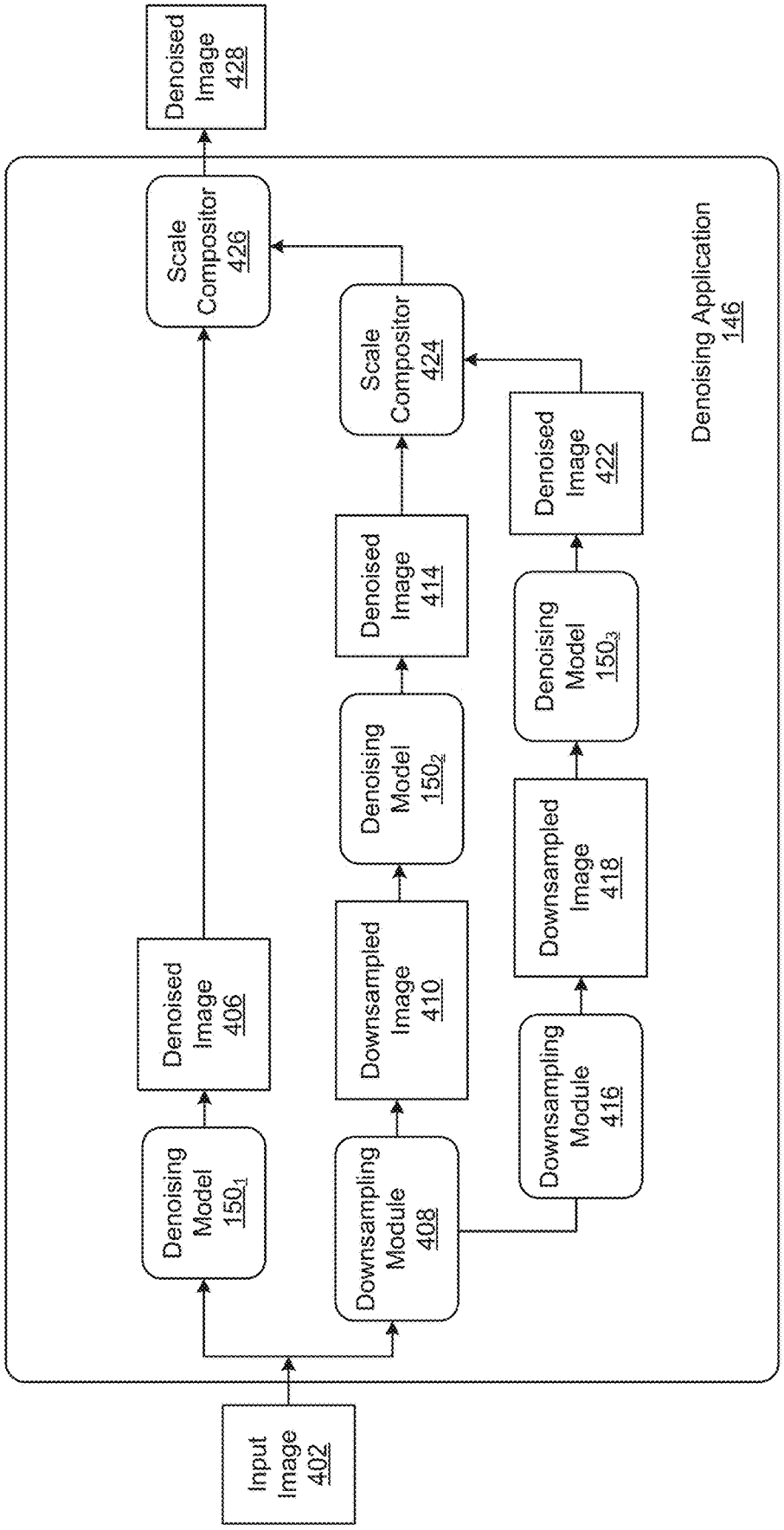

Input Image 402
Denoising Application 146
Denoising Model 150$_1$
Denoised Image 406
Downsampling Module 408
Downsampled Image 410
Denoising Model 150$_2$
Denoised Image 414
Downsampling Module 416
Downsampled Image 418
Denoising Model 150$_3$
Denoised Image 422
Scale Compositor 424
Scale Compositor 426
Denoised Image 428

FIG. 4

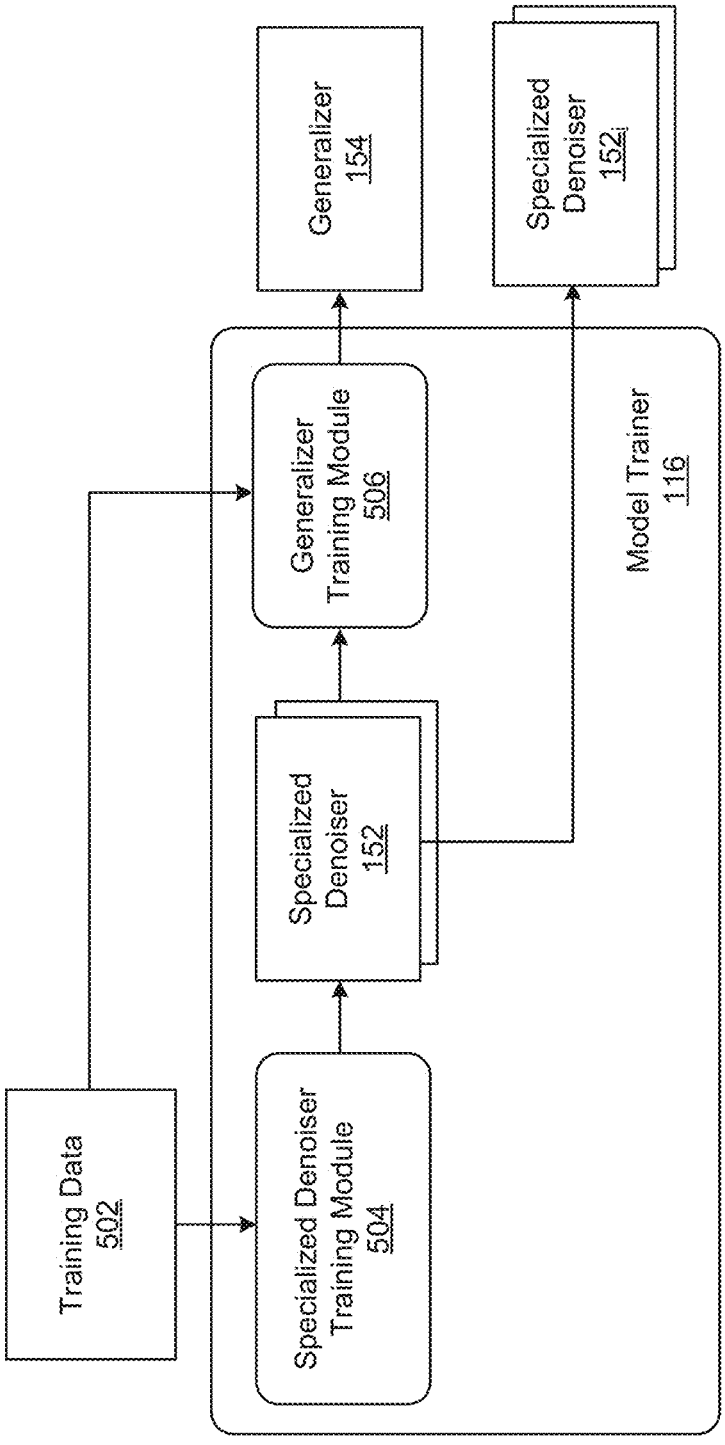

Training Data
502
Specialized Denoiser Training Module
504
Specialized Denoiser
152
Generalizer Training Module
506
Generalizer
154
Specialized Denoiser
152i
Model Trainer
116

FIG. 5

FIG. 6A

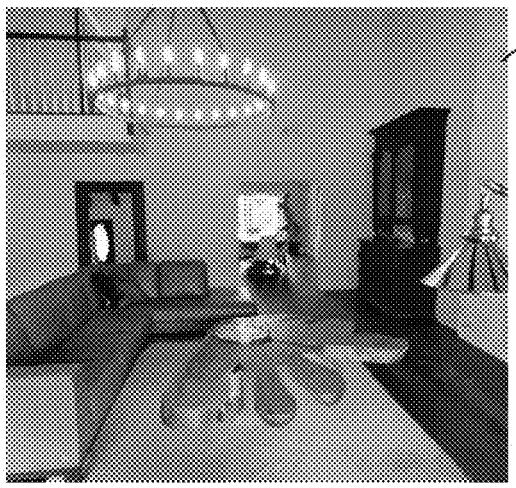


FIG. 6B (Prior Art)

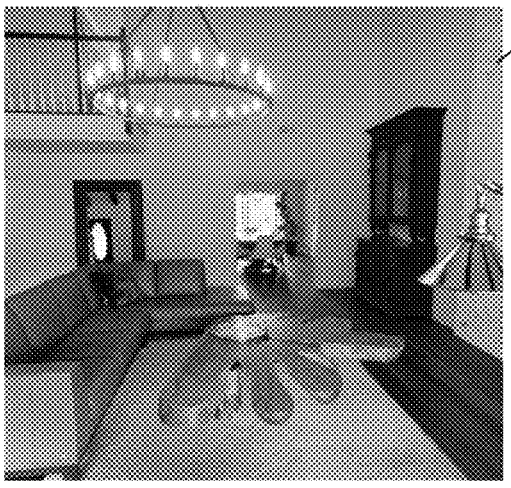


FIG. 6C

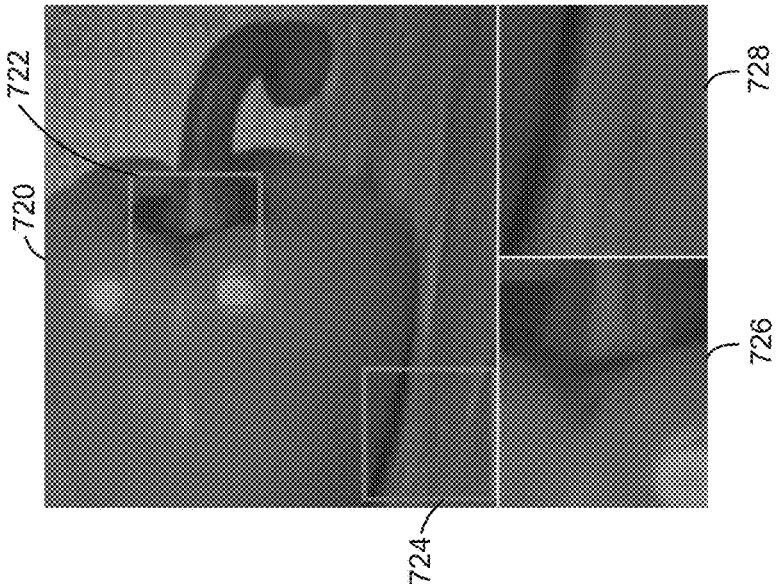


FIG. 7C

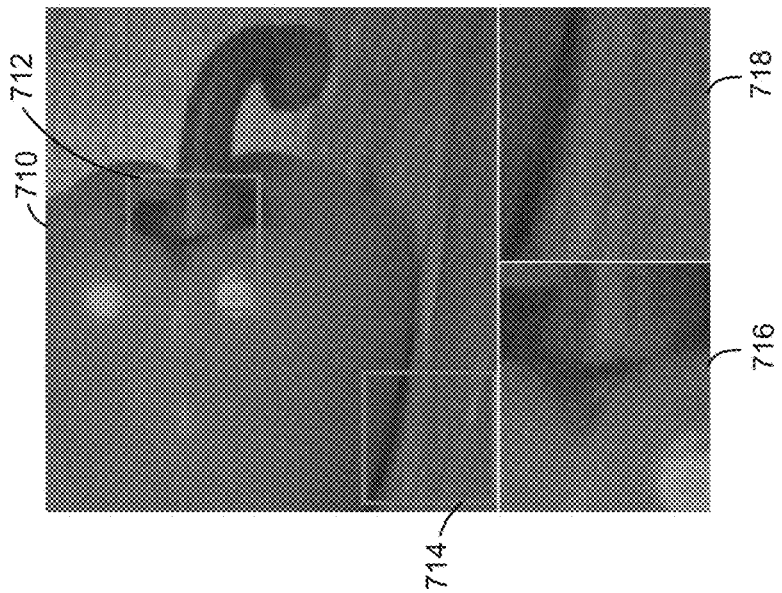


FIG. 7B (Prior Art)

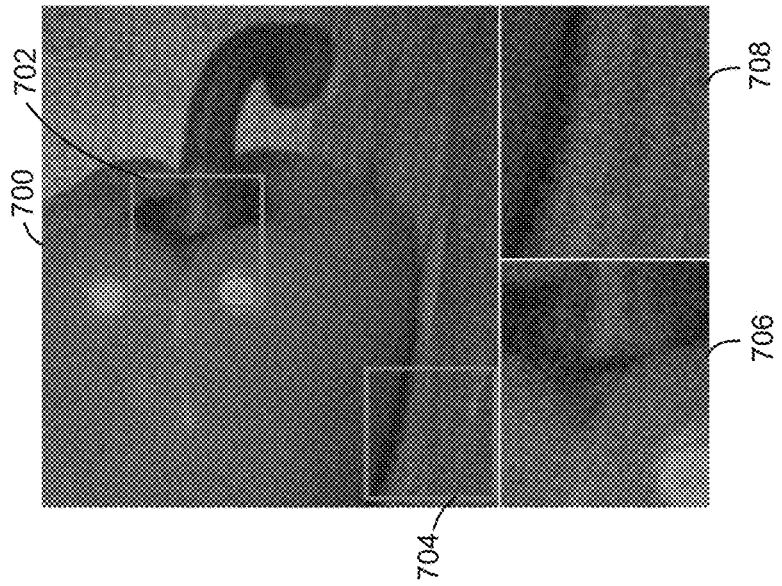


FIG. 7A

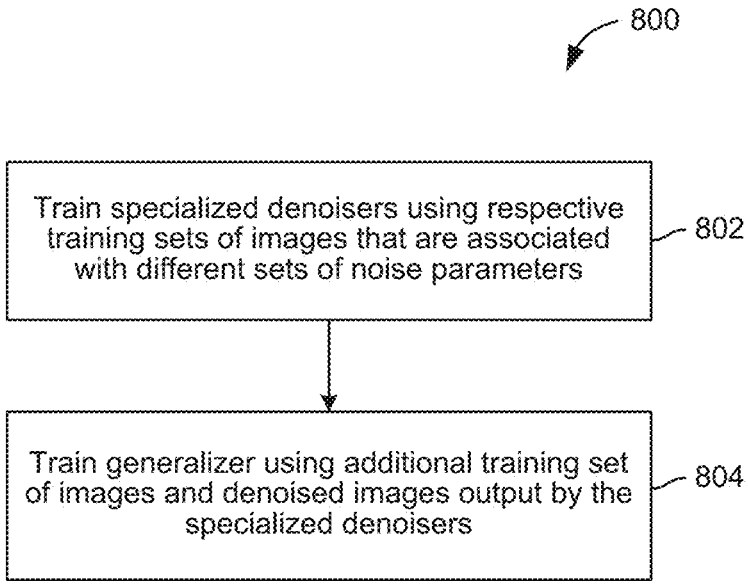

800
Train specialized denoisers using respective training sets of images that are associated with different sets of noise parameters
802
Train generalizer using additional training set of images and denoised images output by the specialized denoisers
804

FIG. 8

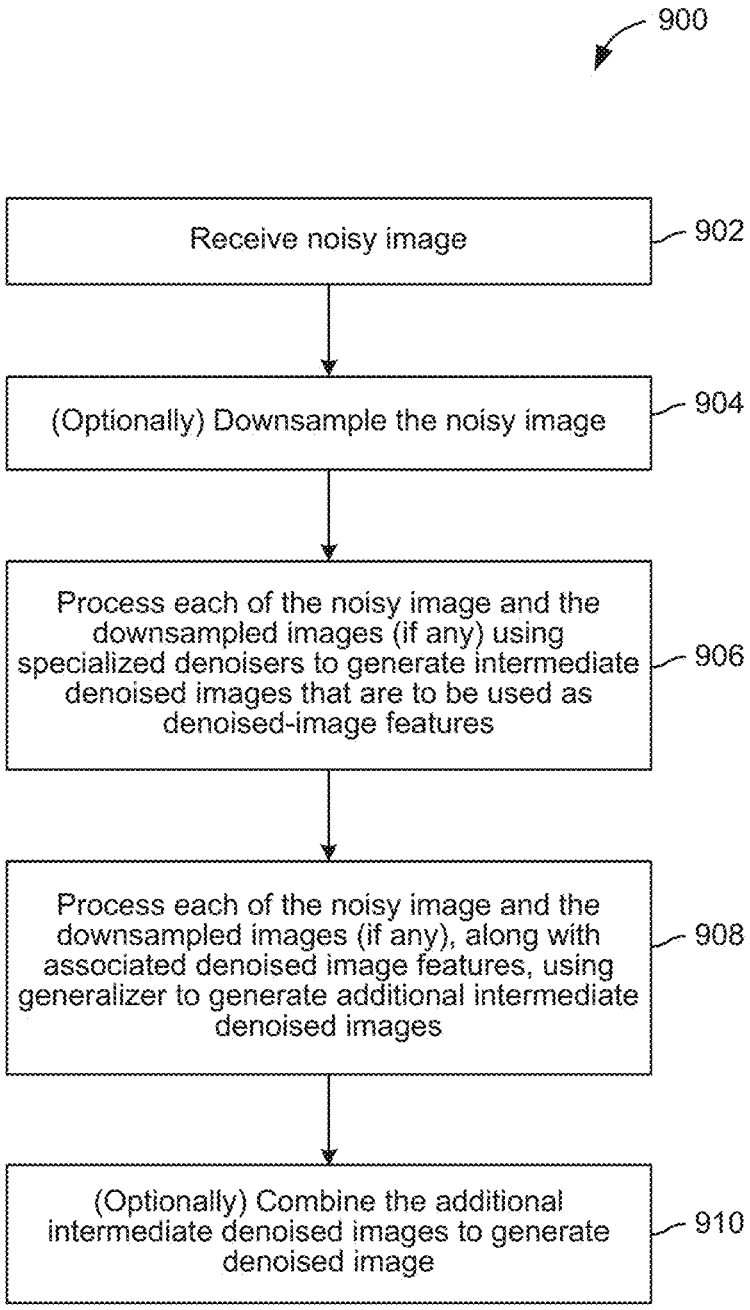

900
Receive noisy image
902
(Optionally) Downsample the noisy image
904
Process each of the noisy image and the downsampled images (if any) using specialized denoisers to generate intermediate denoised images that are to be used as denoised-image features
906
Process each of the noisy image and the downsampled images (if any), along with associated denoised image features, using generalizer to generate additional intermediate denoised images
908
(Optionally) Combine the additional intermediate denoised images to generate denoised image
910

FIG. 9

TECHNIQUES FOR ROBUST IMAGE DENOISING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority benefit of the United States Provisional Patent Application titled, "ROBUST IMAGE DENOISING USING KERNEL PREDICTING NETWORKS," filed on Nov. 15, 2019 and having Ser. No. 62/936,341. The subject matter of this related application is hereby incorporated herein by reference.

BACKGROUND

Technical Field

[0002] Embodiments of the present disclosure relate generally to image processing and denoising and, more specifically, to techniques for robust image denoising.

Description of the Related Art

[0003] Image noise refers to random variation in the color and/or brightness within images. Noise is a typical byproduct in images that are rendered using Monte Carlo rendering techniques. Images (i.e., photographs) captured by cameras can also contain noise due to the mechanics of camera sensors. For example, the quality of images captured in low-lighting conditions can be adversely affected by noise.

[0004] Image denoising is the process of removing noise from an image. Conventional approaches for image denoising utilize either specialized denoisers or "blind" denoisers. Specialized denoisers, such as deep denoising networks, can be trained to remove specific types of noise from images, where the noise is typically within narrow ranges of one or more noise parameters, such as additive white Gaussian noise having a certain magnitude. However, the performance of specialized denoisers drops rapidly for images having noise with noise parameters that are different than the training noise parameters.

[0005] In contrast to specialized denoisers, blind denoisers are typically trained using images having noise with diverse noise parameters. Although blind denoisers can be robust to different noise parameters, such robustness comes at the cost of overall denoising quality. In particular, the denoising quality of blind denoisers is generally lower than specialized denoisers for images having noise with the particular noise parameters that the specialized denoisers were trained to denoise.

[0006] Currently, there are few, if any, denoising techniques that optimally balance or combine the denoising quality of specialized denoisers with the denoising robustness of blind denoisers.

[0007] As the foregoing illustrates, what is needed in the art are more effective techniques for denoising images.

SUMMARY

[0008] One embodiment of the present application sets forth a computer-implemented method for denoising an image. The method includes processing the image using a plurality of first denoisers to generate a plurality of first denoised images, where each first denoiser is trained to denoise images associated with at least one noise parameter. The method further includes processing the image and the plurality of first denoised images using a second denoiser to generate a second denoised image.

[0009] Another embodiment of the present application sets forth a computer-implemented method for training denoisers. The method includes training each first denoiser included in a plurality of first denoisers using a respective set of images associated with at least one noise parameter and ground truth images corresponding to images included in the respective set of images. The method further includes training a second denoiser using an additional set of images, ground truth images corresponding to images included in the additional set of images, and a plurality of denoised images, where the plurality of denoised images is generated by processing the additional set of images using the plurality of first denoisers.

[0010] Other embodiments of the present disclosure include, without limitation, a computer-readable medium including instructions for performing one or more aspects of the disclosed techniques as well as a computing device for performing one or more aspects of the disclosed techniques.

[0011] At least one technical advantage of the disclosed techniques relative to the prior art is that the disclosed techniques bypass the traditional tradeoff between the denoising quality of specialized denoisers and the generalizability and robustness of blind denoisers. In that regard, the disclosed techniques combine the performance of specialized denoisers with the generalizing capabilities of blind denoisers. Experience has shown that the disclosed techniques can achieve better denoising quality than conventional specialized denoisers when applied to images having noise with arbitrary noise parameters, such as different noise magnitudes. The disclosed techniques also can achieve better overall denoising quality than conventional blind denoisers. In addition, the disclosed techniques can denoise images more quickly than many conventional denoisers. These technical advantages represent one or more technological improvements over prior art approaches.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] So that the manner in which the above recited features of the disclosure can be understood in detail, a more particular description of the disclosure, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this disclosure and are therefore not to be considered limiting of its scope, for the disclosure may admit to other equally effective embodiments.

[0013] FIG. **1** illustrates a system configured to implement one or more aspects of the various embodiments;

[0014] FIG. **2** is a more detailed illustration of the denoising model of FIG. **1**, according to various embodiments;

[0015] FIG. **3**A illustrates an exemplar noisy image;

[0016] FIG. **3**B illustrates exemplar features generated by the specialized denoisers of FIG. **1** for the exemplar region of FIG. **3**A, according to various embodiments;

[0017] FIG. **4** is a more detailed illustration of the denoising application of FIG. **1**, according to various embodiments;

[0018] FIG. **5** is a more detailed illustration of the model trainer of FIG. **1**, according to various embodiments;

[0019] FIG. **6**A illustrates another exemplar noisy image;

[0020] FIG. **6**B illustrates a denoised image generated using a conventional blind denoiser, according to the prior art;

[0021] FIG. **6**C illustrates a denoised image generated using the specialized denoisers and the generalizer of FIG. **1**, according to various embodiments;

[0022] FIG. **7**A illustrates another exemplar noisy image;

[0023] FIG. **7**B illustrates a denoised image generated using a conventional specialized denoiser, according to the prior art;

[0024] FIG. **7**C illustrates a denoised image generated using the specialized denoisers and the generalizer of FIG. **1**, according to various embodiments;

[0025] FIG. **8** sets forth a flow diagram of method steps for training multiple specialized denoisers and a generalizer, according to various embodiments; and

[0026] FIG. **9** sets forth a flow diagram of method steps for denoising an image, according to various embodiments.

DETAILED DESCRIPTION

[0027] In the following description, numerous specific details are set forth to provide a more thorough understanding of the present invention. However, it will be apparent to one of skill in the art that embodiments of the present invention may be practiced without one or more of these specific details.

System Overview

[0028] FIG. **1** illustrates a system **100** configured to implement one or more aspects of the various embodiments. As shown, the system **100** includes a machine learning server **110**, a data store **120**, and a computing device **140** in communication over a network **130**, which may be a wide area network (WAN) such as the Internet, a local area network (LAN), or any other suitable network.

[0029] As shown, a model trainer **116** executes on a processor **112** of the machine learning server **110** and is stored in a system memory **114** of the machine learning server **110**. The processor **112** receives user input from input devices, such as a keyboard or a mouse. In operation, the processor **112** is the master processor of the machine learning server **110**, controlling and coordinating operations of other system components. In particular, the processor **112** may issue commands that control the operation of a graphics processing unit (GPU) that incorporates circuitry optimized for graphics and video processing, including, for example, video output circuitry. The GPU may deliver pixels to a display device that may be any conventional cathode ray tube, liquid crystal display, light-emitting diode display, or the like.

[0030] The system memory **114** of the machine learning server **110** stores content, such as software applications and data, for use by the processor **112** and the GPU. The system memory **114** may be any type of memory capable of storing data and software applications, such as a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash ROM), or any suitable combination of the foregoing. In some embodiments, a storage (not shown) may supplement or replace the system memory **114**. The storage may include any number and type of external memories that are accessible to the processor **112** and/or the GPU. For example, and without limitation, the storage may include a Secure Digital Card, an external Flash memory, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing.

[0031] It will be appreciated that the machine learning server **110** shown herein is illustrative and that variations and modifications are possible. For example, the number of processors **112**, the number of GPUs, the number of system memories **114**, and the number of applications included in the system memory **114** may be modified as desired. Further, the connection topology between the various units in FIG. **1** may be modified as desired. In some embodiments, any combination of the processor **112**, the system memory **114**, and a GPU may be replaced with any type of virtual computing system, distributed computing system, or cloud computing environment, such as a public, private, or a hybrid cloud.

[0032] As discussed in greater detail below, the model trainer **116** is configured to train machine learning models, including multiple specialized denoisers $\mathbf{152}_{1\text{-}N}$, which are individually referred to as a specialized denoiser **152** and collectively referred to as specialized denoisers **152**, and a generalizer **154**. Each of the specialized denoisers **152** is trained to denoise images associated with specific ranges of noise parameters (i.e., images having noise with noise parameters that are within the specific ranges). For example, the specific ranges of noise parameters could indicate a particular type of camera sensor or a particular range of Gaussian noise magnitudes. By contrast, the generalizer **154** is trained to generate per-pixel kernels (i.e., individual kernels for each pixel) for denoising images associated with arbitrary noise parameters, as opposed to specific ranges of noise parameters. As described in greater detail below, the generalizer **154** receives as input an image to be denoised as well intermediate denoised images generated by the specialized denoisers **152** after the same image is input into the specialized denoisers **152**. Denoised images output by the specialized denoisers **152** are referred to herein as intermediate denoised images to distinguish them from the denoised images output by an overall denoising model **150** that includes both the specialized denoisers **152** and the generalizer **154**, discussed in greater detail below. The generalizer **154** is an example of a second denoiser (with the specialized denoisers **152** being the first denoisers) that can be trained and applied to denoise images given the images themselves, as well intermediate denoised images output by the specialized denoisers **152**, as input. Any technically feasible second denoiser may be used in other embodiments. For example, the second denoiser may generate outputs other than denoising kernels, such as denoised images, in some embodiments.

[0033] The architectures of the specialized denoisers **152** and the generalizer **154**, as well as techniques for training the same, are discussed in greater detail below. Training data and/or trained machine learning models, including the specialized denoisers **152** and the generalizer **154**, may be stored in the data store **120**. In some embodiments, the data store **120** may include any storage device or devices, such as fixed disc drive(s), flash drive(s), optical storage, network attached storage (NAS), and/or a storage area-network (SAN). Although shown as accessible over the network **130**, in some embodiments the machine learning server **110** may include the data store **120**.

[0034] The trained specialized denoisers **152** and generalizer **154** may be deployed to any suitable applications that

denoise images. Illustratively, a denoising application **146** is stored in a memory **144**, and executes on a processor **142** of the computing device **140**. Components of the computing device **140**, including the memory **144** and the processor **142** may be similar to corresponding components of the machine learning server **110**.

[0035] As shown, the denoising application **146** includes the denoising model **150**, which itself includes the specialized denoisers **152** and the generalizer **154**. As described, each of the specialized denoisers **152** is a machine learning model trained to denoise images associated with a specific range of noise parameters. In some embodiments, the specialized denoisers **152** may be denoising convolutional neural networks (DnCNNs), discussed in greater detail below. In alternative embodiments, the specialized denoisers **152** may be other types of denoisers, such as U-net architecture networks. Any number and type of specialized denoisers **152** may be employed in embodiments. For example, three specialized denoisers may be used in some embodiments. In general, if the parameter space of the noise is more complex, then more specialized denoisers may be used, and vice versa if the parameter space of the noise is simpler. It should be noted that there is a trade-off between denoising performance and the number of specialized denoisers **152**, as adding specialized denoisers **152** generally increases computational costs.

[0036] As described, the generalizer **154** is trained to generate per-pixel denoising kernels for denoising images having noise with arbitrary noise parameters. The generalizer **154** takes as input noisy images, as well as intermediate denoised images generated by the specialized denoisers **152**, which are input into the generalizer **154** as additional features and also referred to herein as "denoised-image features." The noisy image may generally be any image that a user wishes to denoise. Examples of such images include live-action images captured by a camera and images rendered via Monte Carlo rendering techniques. For example, a live-action image could be denoised before visual effects are added. As another example, denoising could be applied to a high dynamic range image, which can include a relatively large amount of noise. As yet another example, denoising could be used to accelerate rendering by denoising an initial noisy image generated via Monte Carlo rendering.

[0037] In some embodiments, the generalizer **154** is a kernel-generating machine learning model, such as a kernel predicting convolutional network (KPCN). As discussed in greater detail below in conjunction with FIGS. **2** and **9**, the denoising application **146** can denoise a noisy image by first processing the noisy image using the specialized denoisers **152** to generate intermediate denoised images. Then, the denoising application **146** inputs the intermediate denoised images generated by the specialized denoisers **152** as features, in addition to the noisy image, into the generalizer **154**. In turn, the generalizer **154** outputs per-pixel denoising kernels, which the denoising application **146** normalizes and applies to the noisy image to generate a denoised image.

[0038] In some embodiments, discussed in greater detail below in conjunction with FIGS. **4** and **9**, multi-scale denoising may be performed to more effectively remove low-frequency components of noise. In such cases, a noisy image is downsampled one or more times to different scales prior to being input into the denoising model **150** that includes the specialized denoisers **152** and the generalizer **154**. Then, the denoised outputs of the denoising model **150** at the different scales are combined to generate a final denoised image. This is in contrast to simply processing the noisy image using the specialized denoisers **152** and then inputting the noisy image and the intermediate denoised images generated by the specialized denoisers **152** into the generalizer **154**, as described above.

[0039] Experience has shown that the denoising model **150** can achieve better denoising quality than conventional specialized denoisers. The denoising model **150** can also generalize better than conventional blind denoisers. In addition, the denoising model **150** can denoise images more quickly than some conventional denoisers. It should also be noted that the denoising model **150** does not require noise parameters to be explicitly estimated or a camera imaging pipeline to be modeled.

[0040] The number of machine learning servers and application servers may be modified as desired. Further, the functionality included in any of the applications may be divided across any number of applications or other software that are stored and execute via any number of devices that are located in any number of physical locations.

Robust Image Denoising

[0041] FIG. **2** is a more detailed illustration of the demising model **150** of FIG. **1**, according to various embodiments. In FIGS. **2** and **4-5**, rectangles with rounded edges represent software components and rectangles with square edges represent data, which may be input into and/or output from the software components.

[0042] As shown, the denoising model **150** is used by the denoising application **146** to process a noisy image **202** and obtain a denoised image **214**. As described, the noisy image **202** may be, e.g., a live-action image captured by a camera or a rendered image. Although described herein primarily with respect to receiving and denoising a single image at a time, in some embodiments, the denoising application **146** may process multiple image frames from a video at once, such as a number of consecutive image frames before and after a given image frame that is being denoised. Doing so allows the denoising application **146** to consider temporal neighborhoods as well as spatial neighborhoods of each pixel.

[0043] The denoising application **146** first inputs the noisy image **202** into the specialized denoisers **152**. As described, the specialized denoisers **152** output intermediate denoised images that are input as additional features $\mathbf{204}_{1\text{-}N}$, along with the noisy image **202**, into a second denoiser, which as shown is the generalizer **154**. Each of the features $\mathbf{204}_{1\text{-}N}$ represents characteristics of the noisy image **202**, namely an intermediate denoised version of the noisy image **202**, that is input into the generalizer **154**. The features $\mathbf{204}_{1\text{-}N}$ are referred to herein individually as a feature **204** and collectively as features **204**. As the features **204** include denoised images, the features **240** are also referred to herein as "denoised-image features." Although described herein with respect to denoised-image features for simplicity, some embodiments may actually employ three channels, namely the red, green, and blue channels, for each of the denoised-image features as well as for the noisy image.

[0044] As described, each of the specialized denoisers **152** is trained to demise images associated with a specific range of noise parameters. The noise parameters could indicate, e.g., a particular type of camera sensor or a particular range of Gaussian noise magnitudes. In some embodiments, each

of the specialized denoisers **152** is a DnCNN. The DnCNN architecture includes blocks of convolutional layers (including, e.g., 3×3 filters and 64 channels) followed by batch normalization and rectified linear unit (Real) activations, as well as a final layer (e.g., a 3×3 layer) that produces an output. In alternative embodiments, the specialized denoisers **152** may be other types of denoisers, such as U-net architecture networks.

[0045] More formally, each of the specialized denoisers **152** can be trained using noisy images having noise with a specific set of noise parameters $\{\lambda_0, \lambda_1, \ldots, \lambda_s\}$, Subsequent to training, the specialized denoisers **152** may generally perform well in denoising images associated with noise parameters similar to those used to train the specialized denoisers **152**. However, the denoising performance of the specialized denoisers **152** drops rapidly for images having noise with different noise parameters.

[0046] To alleviate the inability of the specialized denoisers **152** to generalize over noise parameters, the denoising application **146** further processes the noisy image **202** using the generalizer **154**, The generalizer **154** receives as inputs the noisy image **202** and intermediate denoised images, which are output by the specialized denoisers **152** and input as the features **204** into the generalizer **154**. As shown, the generalizer **154** outputs per-pixel denoising kernels **206**, which are individually referred to as a denoising kernel **206** and collectively referred to as per-pixel denoising kernels **206**. In some embodiments, the generalizer **154** may be a KPCN. For computational efficiency reasons, the number of denoised image features will generally be low, and the noise parameters associated with an image λ_a may not match the noise parameters $\{\lambda_0, \lambda_1, \ldots, \lambda_s\}$ that the specialized denoisers **152** were trained on. The generalizer **154** is therefore used to estimate suitable per-pixel denoising kernels **206** for the noisy image **202** associated with noise parameters λ_a of any input image given the image itself and denoised-image features output by the specialized denoisers **152**. Doing so produces a relatively consistent denoising quality over a wide range of noise parameters, making the denoising model **150** robust to image noise characteristics.

[0047] If the input noisy image **202** is treated as a vector, which can be denoted as $x \in \mathbb{R}^3$, then the denoised-image features are additional channels f, each of which includes an individual feature map $\{f_1, f_2, \ldots, f_s\}$. The generalizer **154** takes as input the tuple $\{x, f\}$. The generalizer **154** is trained to minimize an average distance between estimated denoised images $\hat{x}$ and corresponding noise-free ground truth images y in a training data set. The noise-free ground truth images in the training data set are also referred to herein as "clean" images and may be images to which noise (e.g., Gaussian noise) has been applied to generate noisy images. It should be understood that the training data may generally include such noisy images as well as the corresponding clean images. More formally, the generalizer **154** may be expressed as $\hat{x}=(d\{x, f\}; \hat{\theta})$, where d denotes a denoiser with parameters $\hat{\theta}$. During training, the parameters $\hat{\theta}$ are determined in a supervised setting using the dataset $\{\{x^1, y^1\}, \{x^2, y^2\}, \ldots, \{x^n, y^n\}\}$, with the objective, or loss function, being:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} l(y, d(\{x, f\}; \theta)). \quad (1)$$

[0048] The trained generalizer **154** can be used to estimate a k×k kernel of scalar weights around the neighborhood $\mathcal{N}(p)$ of a pixel location p, which is also referred to herein as the denoising kernel **206** for the pixel location p. The denoising application **146** may then use such a denoising kernel **206** to compute a weighted linear combination of the pixels in the neighborhood $\mathcal{N}(p)$ to obtain the final pixel color prediction at the pixel location p. As shown, a normalization module **208** of the denoising application **146** then normalizes each of the per-pixel denoising kernels **206** to generate normalized kernels **210**, after which a weighted reconstruction module **212** of the denoising application **212** applies the normalized kernels **210** to denoise the noisy image **202** via weighted reconstruction. In some embodiments, the denoising application **146** normalizes weights of the per-pixel denoising kernels **206** using a softmax function so that the weights have values within [0,1]. The denoised image **214** is then output from the denoising model **150**.

[0049] As described, the generalizer **154** is a KPCN in some embodiments. The KPCN architecture includes residual blocks, each of which can include 3×3 convolution layers bypassed by a skip connection. As described, rather than directly predicting pixel colors, KPCNs predict per-pixel denoising kernels, which can provide significant improvement in convergence speed during training. For example, the KPCN could output a 5×5 denoising kernel for each pixel. In some embodiments, the KPCN generalizer **154** estimates a denoising kernel at pixel location p using the following identity:

$$\hat{x}_p = d_p(\{x_f\}; \hat{\theta}) = \sum_{q \in \mathcal{N}(p)} w_{pq} x_q, \quad (2)$$

where w_{pq} denotes a normalized estimated weight at location q belonging to the kernel at pixel location p.

[0050] FIGS. **3**A-**3**B illustrate an exemplar noisy image and denoised-image features generated by the specialized denoisers **152**, according to various embodiments. As shown in FIG. **3**A, a noisy image **300** depicts a wooden floor. FIG. **3**B shows intermediate denoised image outputs **310**, **312**, and **314** generated by three specialized denoisers **152** given the noisy image **300** as input. As described, such intermediate denoised images **310**, **312**, and **314** can be input as additional denoised-image features into the generalizer **154**.

[0051] As the specialized denoisers **152** are trained for particular noise parameters, such as a specific noise type (e.g., noise produced by a specific type of camera sensor) or a range of noise parameters (e.g., additive Gaussian noise with a specific magnitude), the quality of each intermediate denoised image **310**, **312**, and **314** will vary depending on the noise parameters associated with the input noisy image **300**. As described, the generalizer **154** can be used to improve the final denoising quality by taking as input the intermediate denoised images **310**, **312**, and **314** as features in addition to the noisy image **300**.

[0052] FIG. **4** is a more detailed illustration of the denoising application **146** of FIG. **1**, according to various embodiments. This illustration assumes that multi-scale denoising is performed to more effectively remove low-frequency components of noise. As described, such multi-scale denoising is optional, and, in other embodiments, the denoising application **146** may simply process a noisy image using the specialized denoisers **152** and then process the noisy image and intermediate denoised images generated by the specialized denoisers **152** using the generalizer **154** to obtain per-pixel denoising kernels that can be used to denoise the original, noisy image.

[0053] As shown in FIG. **4**, downsampling modules **408** and **416** of the denoising application **146** downsample an input image **402** to generate downsampled images **410** and **418**, respectively. In some embodiments, the input image **402** is uniformly downsampled to half the original size by the downsampling module **408**, and then downsampled again to half the previously downsampled size (i.e., to a quarter of the original size) by the downsampling module **416**. Although one input image **402** is shown for illustrative purposes, in some embodiments, the denoising application **146** may process multiple image frames from a video, such as a number of consecutive image frames before and after a given image frame that is being denoised.

[0054] As shown, the denoising application separately inputs the image **402** and the downsampled images **410** and **418** into the denoising models $\mathbf{150}_{1\text{-}3}$, to obtain an intermediate denoised image **406**, an intermediate denoised image **414**, and an intermediate denoised image **422**, respectively. Each of the denoising models $\mathbf{150}_{1\text{-}3}$ shown in FIG. **4** is the same as the denoising model **150**, and processing of the image **402** and the downsampled images **410** and **418** using the denoising models $\mathbf{150}_{1\text{-}3}$ may either be performed in parallel or serially. As described, the generalizer **154** of the denoising model **150** outputs, for each pixel location, a respective kernel (e.g., a fixed-size 5×5 kernel). By determining such per-pixel kernels at different scales obtained via downsampling, the effective kernel size is increased. Doing so increases the perceptive field and helps remove low-frequency components of noise, which can be more challenging to remove and require larger filter kernels.

[0055] The intermediate denoised images **406**, **414**, and **422** are combined, starting from a combination of the coarsest denoised image **422** with the finer denoised image **414** using a scale compositor **424** and proceeding to a combination of an output of the scale compositor **424** with the finest denoised image **406** using a scale compositor **426**. The output of the scale compositor **426** is a denoised image **428** that is output by the denoising application **145**. In some embodiments, each of the scale compositors **424** and **426** takes as input a pair of coarse and fine scale images (i^c and i^f), such as the intermediate denoised image **422** and the intermediate denoised image **414** or the output of the scale compositor **424** and the intermediate denoised image **406**, as shown in FIG. **4**. In such cases, each of the scale compositors **424** and **426** extracts, for each pixel p, a scalar weight α_p that is used to blend consecutive scales to generate a blended image as follows:

$$o_p^f = i_p^f - \alpha_p (UDi^f)_p + \alpha_p (Ui^c)_p, \tag{3}$$

where D denotes downsampling (e.g., a 2×2 downsampling) and U denotes upsampling (e.g., a 2×2 upsampling). Such a blending replaces low frequencies of the fine scale image i^f with low frequencies obtained from the coarse scale image i^c and produces the output o^f. The scalar weight α_p balances low frequencies from the finer and coarser scales. In some embodiments, the scalar weight α_p may be determined by taking the intermediate images produced by denoising two adjacent scales of a frame (or a sequence), namely a coarse-scale image i^c and a fine-scale image i^f, and inputting those intermediate denoised images into a convolutional network that extracts a per-pixel scalar weight α_p, which can then be used to blend the coarse- and fine-scale intermediate denoised images according to equation (3). Although described herein primarily with respect to successive scale compositors that each combine a pair of coarse and fine scale intermediate denoised images, in alternative embodiments a single scale compositor that combines intermediate denoised images at all of the different scales (e.g., the denoised images **406**, **414**, and **422**) may be used.

[0056] FIG. **5** is a more detailed illustration of the model trainer **116** of FIG. **1**, according to various embodiments. As shown, a two-step training procedure is employed in which the model trainer **116** uses training data **502** that includes a set of noisy images and corresponding clean images to first train the specialized denoisers **152**. Then, the model trainer **116** uses the training data **502** and the trained specialized denoisers **152** to train the generalizer **154**. As described, each of the specialized denoisers **152** is trained using noisy images having noise with a different set of noise parameters and corresponding clean images. Accordingly, the training of each of the specialized denoisers **152** at **504** will generally only use some of the noisy images (and corresponding clean images) in the training data **502** having noise with the particular set of noise parameters for which the specialized denoiser **152** is being trained. By contrast, a generalizer training module **506** can use a more diverse set of images to train the generalizer **154**, such as all of the noisy images in the training data **502** and corresponding clean images, as well as intermediate denoised images output by the specialized denoisers **152** when given the noisy images from the training data **502**. For example, Gaussian noise of random sigma ranging from **10** to **40** could be added to each image in a data set, and such noisy images as well as intermediate denoised images that are generated by the specialized denoiser **152** can be fed into the generalizer **154** during training.

[0057] Embodiments may utilize any technically feasible training techniques to train the specialized denoisers **152** and the generalizer **154**. In some embodiments, both the generalizer **154** and the specialized denoisers **152** are trained using image patches of size 128×128, using mini-batch size 16 and the Adam optimizer with an initial learning rate of 10^{-4}, as well as dataset specific schedulers to decay the learning rate during the course of training. In some embodiments, the training may use the mean absolute percentage error (MAPE) to assess the distance to a clean reference in the training data **502**:

$$l(y, \hat{x}) = \frac{|y - \hat{x}|}{|y| + \epsilon}, \tag{4}$$

where $\hat{x}$ is the denoised image, y is the clean reference image, and $\epsilon = 10^{-3}$ is used to avoid division by zero. Other loss functions, such as L_1, root mean squared error (RMSE), and structural similarity (SSIM) may be used in alternative embodiments.

[0058] FIGS. **6A-6C** illustrate an exemplar noisy image **600**, a denoised image **602** generated using a conventional blind denoiser, and a denoised image **604** generated using the specialized denoisers **152** and the generalizer **154** of FIG. **1**, according to various embodiments. Illustratively, the noisy image **600** is a live-action image. As described, live-action images can be denoised before visual effects are added and in the case of high dynamic range images that include a relatively large amount of noise, among other things. Although a live-action image is shown for illustrative

purposes, techniques disclosed herein can also be applied to denoise rendered images, such as those generated using Monte Carlo rendering techniques. In contrast to rendered images, no additional information (e.g., albedo, surface normal, depth, etc.) beyond colors are typically available for live-action images that a denoiser can take advantage of.

[0059] As shown in FIG. **6**C, the denoised image **604** generated from the noisy image **600** using the specialized denoisers **152** and the generalizer **154** has higher quality than the denoised image **602** generated using the conventional blind denoiser, shown in FIG. **6**B. Quality may be measured as, e.g., the average distance from a corresponding clean image. Experience has shown that the specialized denoisers **152** and the generalizer **154** can generate denoised images that are sharper and more robust to varying levels of noise magnitudes compared to conventional blind denoisers. In addition, denoised images generated using the specialized denoisers **152** and the generalizer **154** can be at least as good as specialized denoisers for images having noise with noise parameters that the specialized denoisers were trained to denoise.

[0060] FIGS. **7**A-**7**C illustrate another exemplar noisy image **700**, a denoised image **710** generated using a conventional specialized denoiser, and a denoised image **720** generated using the specialized denoisers **152** and the generalizer **154** of FIG. **1**, according to various embodiments. Zoomed-in views **706** and **708** of regions **702** and **704**, respectively, within the noisy image are shown in FIG. **7**A, Likewise, zoomed-in views **716** and **718** of regions **712** and **714**, respectively, within the denoised image **710** are shown in FIG. **7**B, and zoomed-in views **706** and **708** of regions **702** and **704**, respectively, within the denoised image **720** are shown in FIG. **7**C.

[0061] Similar to the noisy image **600**, the noisy image **700** is shown as a live-action image, but may alternatively be a rendered image. As shown in FIG. **7**C, the denoised image **720** generated from the noisy image **700** using the specialized denoisers **152** and the generalizer **154** has higher quality than the denoised image **710** generated using the conventional specialized denoiser, shown in FIG. **7**B. In generating the denoised image **710**, a specialized U-Net was used as the conventional specialized denoiser. In generating the denoised image **720**, five U-Nets and a KPCN were used as the specialized denoisers **152** and the generalizer **154**, respectively.

[0062] FIG. **8** sets forth a flow diagram of method steps for training the specialized denoisers **152** and the generalizer **154**, according to various embodiments. Although the method steps are described in conjunction with the system of FIG. **1**, persons of ordinary skill in the art will understand that any system configured to perform the method steps, in any order, is within the scope of the present disclosure.

[0063] As shown, a method **800** begins at step **802**, where the model trainer **116** trains the specialized denoisers **152** using respective training sets of images that are associated with different sets of noise parameters. As described, each of the specialized denoisers **152** is trained to demise images associated with a specific range of noise parameters.

[0064] At step **804**, the model trainer **116** trains the generalizer **154** using an additional training set of images and denoised images output by the specialized denoisers **152** after the additional set of images is input into the trained specialized denoisers **152**. As described, the generalizer **154** is trained to take as inputs a noisy image and intermediate denoised images generated by the specialized denoisers **152** given the same noisy image, and to output per-pixel denoising kernels. In some embodiments, the generalizer **154** is trained using a more diverse set of images than the sets of images used to train each of the specialized denoisers **152** individually.

[0065] FIG. **9** sets forth a flow diagram of method steps for denoising an image, according to various embodiments. Although the method steps are described in conjunction with the system of FIG. **1**, persons of ordinary skill in the art will understand that any system configured to perform the method steps, in any order, is within the scope of the present disclosure.

[0066] As shown, a method **900** begins at step **902**, where the denoising application **146** receives a noisy image. The noisy image may be, e.g., a live-action image or a rendered image. Further, the noisy image may either be a stand-alone image or an image frame within a video. In some embodiments, the denoising application **146** may receive and process other images in addition to the noisy image, such as a number of consecutive image frames before and after the noisy image, as discussed above in conjunction with FIGS. **2** and **4**.

[0067] At step **904**, the denoising application **146** optionally downsamples the noisy image. As described, downsampling may be performed in the multi-scale case to increase the perceptive field and remove low-frequency components of noise. In some embodiments, the noisy image may be uniformly downsampled to half the original size during each of one or more downsampling operations.

[0068] At step **906**, the denoising application **146** processes each of the noisy image and the downsampled images (if any) using the specialized denoisers **152** to generate intermediate denoised images that are to be used as denoised-image features. As described, each of the specialized denoisers **152** is trained to denoise images associated with a specific range of noise parameters. Intermediate denoised images output by the specialized denoisers **152** are then input as additional features, along with the original noisy image, into the generalizer **154**.

[0069] At step **908**, the denoising application **146** processes each of the noisy image and the downsampled images (if any), along with associated denoised-image features, using the generalizer **154** to generate additional intermediate denoised images. As described, the generalizer **154** is a kernel predicting network, such as a KPCN, in some embodiments. In such cases, the denoising application **146** processes each of the noisy image and the downsampled images (if any), along with associated denoised-image features, using the kernel predicting network to obtain respective per-pixel denoising kernels. Then, the denoising application **146** applies the respective denoising kernels to the noisy image and the downsampled images (if any) to generate the additional intermediate denoised images at step **908**. As described, other embodiments may employ a second denoiser other than the generalizer **154** that is trained and applied to denoise images given the images themselves, as well intermediate denoised images output by the specialized denoisers **152**, as input

[0070] If the optional downsampling at step **904** is not performed, then the denoised image generated by processing the noisy image at step **908** is the final denoised image that is output by the denoising application **146**. Otherwise, the method **900** proceeds to step **910**.

[0071] At step **910**, the denoising application **146** optionally combines the intermediate denoised images to generate a denoised image. This step assumes that the denoising application **146** downsampled the noisy image at step **904**. As described, in some embodiments, the denoising application **146** combines the intermediate denoised images using scale compositors, beginning with a coarsest intermediate denoised image and proceeding to finer intermediate denoised images.

[0072] In sum, techniques are disclosed for training and applying a denoising model that is capable of removing noise from images where attributes of the noise are not known a priori. The denoising model includes multiple specialized denoisers and a generalizer, each of which is a machine learning model. The specialized denoisers are trained to denoise images associated with specific ranges of noise parameters. The generalizer is trained to generate per-pixel denoising kernels for denoising images associated with arbitrary noise parameters using outputs of the trained specialized denoisers. Subsequent to training, a noisy image, such as a live-action image or a rendered image, can be denoised by inputting the noisy image into the specialized denoisers to obtain intermediate denoised images that are then input, along with the noisy image, into the generalizer to obtain per-pixel denoising kernels, which can be normalized and applied to denoise the noisy image.

[0073] At least one technical advantage of the disclosed techniques relative to the prior art is that the disclosed techniques bypass the traditional tradeoff between the denoising quality of specialized denoisers and the generalizability and robustness of blind denoisers. In that regard, the disclosed techniques combine the performance of specialized denoisers with the generalizing capabilities of blind denoisers. Experience has shown that the disclosed techniques can achieve better denoising quality than conventional specialized denoisers when applied to images having noise with arbitrary noise parameters, such as different noise magnitudes. The disclosed techniques also can achieve better overall denoising quality than conventional blind denoisers. In addition, the disclosed techniques can denoise images more quickly than many conventional denoisers. These technical advantages represent one or more technological improvements over prior art approaches.

[0074] Any and all combinations of any of the claim elements recited in any of the claims and/or any elements described in this application, in any fashion, fall within the contemplated scope of the present invention and protection.

[0075] 1. In some embodiments, a computer-implemented method for denoising an image comprises processing the image using a plurality of first denoisers to generate a plurality of first denoised images, wherein each first denoiser is trained to denoise images associated with at least one noise parameter, and processing the image and the plurality of first denoised images using a second denoiser to generate a second denoised image.

[0076] 2. The computer-implemented method of clause 1, further comprising downsampling the image to generate a downsampled image, processing the downsampled image using the plurality of first denoisers to generate a plurality of third denoised images, processing the downsampled image and the plurality of third denoised images using the second denoiser to generate a fourth denoised image, and combining the second denoised image and the fourth denoised image to produce a first blended image.

[0077] 3, The computer-implemented method of clauses 1 or 2, wherein the second denoised image and the fourth denoised image are combined using a scale compositor that receives the second denoised image and the fourth denoised image as input and outputs the first blended image.

[0078] 4. The computer-implemented method of any of clauses 1-3, wherein the second denoiser comprises a kernel predicting convolutional network (KPCN), and processing the image and the plurality of first denoised images using the second denoiser comprises inputting the image and the plurality of first denoised images into the second denoiser to determine per-pixel kernels, normalizing the per-pixel kernels to generate normalized kernels, and denoising the image using the normalized kernels to generate the second denoised image.

[0079] 5. The computer-implemented method of any of clauses 1-4, wherein each first denoiser included in the plurality of first denoisers comprises a denoising convolutional neural network (DnCNN).

[0080] 6. The computer-implemented method of any of clauses 1-5, wherein each first denoiser included in the plurality of first denoisers comprises a U-net architecture network.

[0081] 7, The computer-implemented method of any of clauses 1-6, wherein the at least one noise parameter indicates at least one of a type of camera sensor or a range of noise magnitudes.

[0082] 8. The computer-implemented method of any of clauses 1-7, wherein the image is captured by a camera.

[0083] 9. The computer-implemented method of any of clauses 1-8, wherein the image comprises a rendered image.

[0084] 10. In some embodiments, a computer-implemented method for training denoisers comprises training each first denoiser included in a plurality of first denoisers using a respective set of images associated with at least one noise parameter and ground truth images corresponding to images included in the respective set of images, and training a second denoiser using an additional set of images, ground truth images corresponding to images included in the additional set of images, and a plurality of denoised images, wherein the plurality of denoised images is generated by processing the additional set of images using the plurality of first denoisers.

[0085] 11. The computer-implemented method of clause 10, wherein each first denoiser included in the plurality of first denoisers and the second denoiser is trained using a mean absolute percentage error loss function.

[0086] 12. The computer-implemented method of clauses 10 or 11, wherein the at least one noise parameter indicates at least one of a type of camera sensor or a range of noise magnitudes.

[0087] 13. The computer-implemented method of any of clauses 10-12, wherein each first denoiser included in the plurality of first denoisers and the second denoiser receives input comprising a single image to be denoised.

[0088] 14, The computer-implemented method of any of clauses 10-13, wherein each first denoiser included in the plurality of first denoisers and the second denoiser receives input comprising an image frame to be denoised and a plurality of image frames before and after the image frame in a video.

[0089] 15. The computer-implemented method of any of clauses 10-14, wherein the second denoiser comprises a kernel predicting convolutional network (KPCN) that takes

as input an image and denoised images generated by processing the image using the plurality of first denoisers.

[0090] 16. The computer-implemented method of any of clauses 10-15, wherein each first denoiser included in the plurality of first denoisers comprises one of a denoising convolutional neural network (DnCNN) or a U-net architecture network.

[0091] 17. The computer-implemented method of any of clauses 10-16, wherein each image of the respective set of images and the additional set of images comprises a rendered image or an image captured by a camera.

[0092] 18. In some embodiments, a computer-readable storage medium includes instructions that, when executed by a processing unit, cause the processing unit to perform steps for denoising an image, the steps comprising processing the image using a plurality of first denoisers to generate a plurality of first denoised images, wherein each first denoiser is trained to denoise images associated with at least one noise parameter, and processing the image and the plurality of first denoised images using a second denoiser to generate a second denoised image.

[0093] 19. The computer-readable storage medium of clause 18, the steps further comprising downsampling the image to generate a downsampled image, processing the downsampled image using the plurality of first denoisers to generate a plurality of third denoised images, processing the downsampled image and the plurality of third denoised images using the second denoiser to generate a fourth denoised image, and combining the second denoised image and the fourth denoised image to produce a first blended image.

[0094] 20. The computer-readable storage medium of clauses 18 or 19, wherein the second denoiser comprises a kernel predicting convolutional network (KPCN), and processing the image and the plurality of first denoised images using the second denoiser comprises inputting the image and the plurality of first denoised images into the second denoiser to determine per-pixel kernels, normalizing the per-pixel kernels to generate normalized kernels, and denoising the image using the normalized kernels to generate the second denoised image.

[0095] The descriptions of the various embodiments have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments.

[0096] Aspects of the present embodiments may be embodied as a system, method or computer program product. Accordingly, aspects of the present disclosure may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "module" or "system," Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[0097] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0098] Aspects of the present disclosure are described above with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the disclosure. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, enable the implementation of the functions/acts specified in the flowchart and/or block diagram block or blocks. Such processors may be, without limitation, general purpose processors, special-purpose processors, application-specific processors, or field-programmable.

[0099] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present disclosure. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function (s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0100] While the preceding is directed to embodiments of the present disclosure, other and further embodiments of the disclosure may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

What is claimed is:

1. A computer-implemented method for denoising an image, the method comprising:

processing the image using a plurality of first denoisers to generate a plurality of first denoised images; wherein each first denoiser is trained to denoise images associated with at least one noise parameter; and

processing the image and the plurality of first denoised images using a second denoiser to generate a second denoised image.

2. The computer-implemented method of claim **1**, further comprising:

downsampling the image to generate a downsampled image;

processing the downsampled image using the plurality of first denoisers to generate a plurality of third denoised images;

processing the downsampled image and the plurality of third denoised images using the second denoiser to generate a fourth denoised image; and

combining the second denoised image and the fourth denoised image to produce a first blended image.

3. The computer-implemented method of claim **2**, wherein the second denoised image and the fourth denoised image are combined using a scale compositor that receives the second denoised image and the fourth denoised image as input and outputs the first blended image.

4. The computer-implemented method of claim **1**, wherein the second denoiser comprises a kernel predicting convolutional network (KPCN), and processing the image and the plurality of first denoised images using the second denoiser comprises;

inputting the image and the plurality of first denoised images into the second denoiser to determine per-pixel kernels;

normalizing the per-pixel kernels to generate normalized kernels; and

denoising the image using the normalized kernels to generate the second denoised image.

5. The computer-implemented method of claim **1**, wherein each first denoiser included in the plurality of first denoisers comprises a demising convolutional neural network (DnCNN).

6. The computer-implemented method of claim **1**, wherein each first denoiser included in the plurality of first denoisers comprises a U-net architecture network.

7. The computer-implemented method of claim **1**, wherein the at least one noise parameter indicates at least one of a type of camera sensor or a range of noise magnitudes.

8. The computer-implemented method of claim **1**, wherein the image is captured by a camera.

9. The computer-implemented method of claim **1**, herein the image comprises a rendered image.

10. A computer-implemented method for training denoisers, the method comprising:

training each first denoiser included in a plurality of first denoisers using a respective set of images associated with at least one noise parameter and ground truth images corresponding to images included in the respective set of images; and

training a second denoiser using an additional set of images, ground truth images corresponding to images included in the additional set of images, and a plurality of denoised images, wherein the plurality of denoised images is generated by processing the additional set of images using the plurality of first denoisers.

11. The computer-implemented method of claim **10**, wherein each first denoiser included in the plurality of first denoisers and the second denoiser is trained using a mean absolute percentage error loss function.

12. The computer-implemented method of claim **10**, wherein the at least one noise parameter indicates at least one of a type of camera sensor or a range of noise magnitudes.

13. The computer-implemented method of claim **10**, wherein each first denoiser included in the plurality of first denoisers and the second denoiser receives input comprising a single image to be denoised.

14. The computer-implemented method of claim **10**, wherein each first denoiser included in the plurality of first denoisers and the second denoiser receives input comprising an image frame to be denoised and a plurality of image frames before and after the image frame in a video.

15. The computer-implemented method of claim **10**, wherein the second denoiser comprises a kernel predicting convolutional network (KPCN) that takes as input an image and denoised images generated by processing the image using the plurality of first denoisers.

16. The computer-implemented method of claim **10**, wherein each first denoiser included in the plurality of first denoisers comprises one of a denoising convolutional neural network (DnCNN) or a U-net architecture network.

17. The computer-implemented method of claim **10**, wherein each image of the respective set of images and the additional set of images comprises a rendered image or an image captured by a camera.

18. A computer-readable storage medium including instructions that, when executed by a processing unit, cause the processing unit to perform steps for denoising an image, the steps comprising:

processing the image using a plurality of first denoisers to generate a plurality of first denoised images, wherein each first denoiser is trained to denoise images associated with at least one noise parameter; and

processing the image and the plurality of first denoised images using a second denoiser to generate a second denoised image.

19. The computer-readable storage medium of claim **18**, the steps further comprising:

downsampling the image to generate a downsampled image;

processing the downsampled image using the plurality of first denoisers to generate a plurality of third denoised images;

processing the downsampled image and the plurality of third denoised images using the second denoiser to generate a fourth denoised image; and

combining the second denoised image and the fourth denoised image to produce a first blended image.

20. The computer-readable storage medium of claim **18**, wherein the second denoiser comprises a kernel predicting convolutional network (KPCN), and processing the image and the plurality of first denoised images using the second denoiser comprises:

inputting the image and the plurality of first denoised images into the second denoiser to determine per-pixel kernels;

normalizing the per-pixel kernels to generate normalized kernels; and

denoising the image using the normalized kernels to generate the second denoised image.

* * * * *